Traffic Sign Detection & Recognition

Riccardo Roscica

Digital Image Processing Course
Prof. Andrea Guerriero
Politecnico di Bari
Master's Degree in Computer Engineering

Academic Year 2024/2025

Context and Importance

Why is traffic sign recognition fundamental?

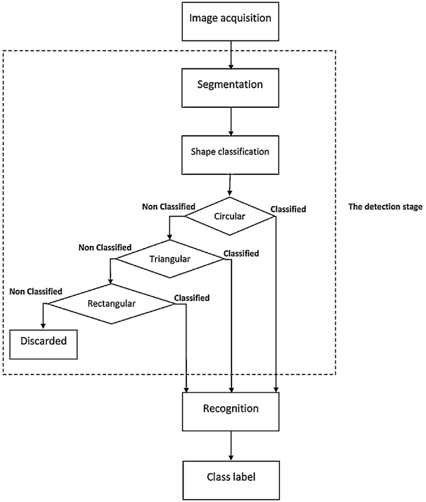
- **ADAS Systems and Autonomous Driving**:
  - Enabling the operation of autonomous vehicles.
  - Providing active driving assistance and compliance with traffic rules.
- **Traffic Management and Smart Cities**:
  - Automatic monitoring of traffic sign compliance.
  - Real-time analysis of vehicle flow.
  - Integration into intelligent mobility systems for optimal traffic management.

Related Work: Ellahyani et al.

Key points of Ellahyani et al.'s work:

- **Color Segmentation:**
  - Use of the **HSI** color space to isolate signs from the background.
  - Application of morphological operations to improve segmentation.
- **Shape Classification:**
  - Use of **Hu moments** to identify the geometric shapes of signs.
- **Sign Recognition:**
  - Extraction of **HOG** and **LSS** features.
  - Final classification using a **Random Forest** model.

Ellahyani et al. Pipeline



Dataset Used: GTSRB

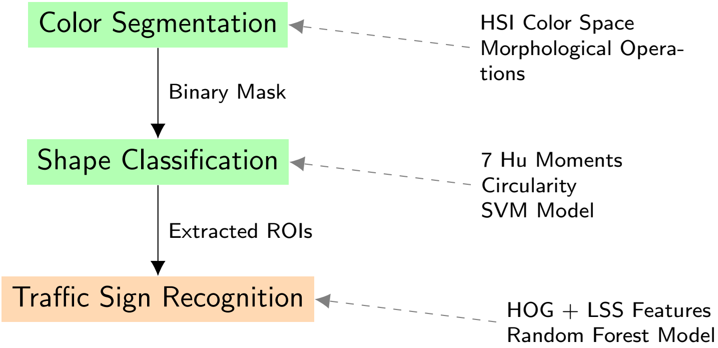
German Traffic Sign Recognition Benchmark (GTSRB)

- **Classes:** 43 different categories of traffic signs.
- **Training images:** 39,209.
- **Testing images:** 12,630.

Usage:

- Training the SVM model for shape classification.
- Training the Random Forest model for sign recognition.

Proposed Pipeline



Pipeline Testing:

- Real-world images from Google Maps Street View for end-to-end validation.
- Verification of detection phases (segmentation and shape classification) and recognition (feature extraction and sign classification).

Color Segmentation

Objective: Isolate red and blue pixels to identify traffic signs.

Choice of Color Space: HSI (Hue, Saturation, Intensity)

- **Advantages:**
  - Robustness to illumination variations.
  - Reduced noise compared to the RGB space.

Thresholds for Color Segmentation

The thresholds defined for each channel of the HSI space allow the isolation of colors of interest.

Color	**Hue**	**Saturation**	**Intensity**
Red	$0 \leq H \leq 10$ **or** $300 \leq H \leq 360$	$25 \leq S \leq 250$	$30 \leq I \leq 200$
Blue	$190 \leq H \leq 260$	$70 \leq S \leq 250$	$56 \leq I \leq 128$

Table: Thresholds defined for red and blue colors.

Creation of the Binary Mask

Process:

- Application of thresholds to obtain a binary mask highlighting red and blue pixels.
- **Noise Removal:**
  - `imfill`: Filling holes in segmented regions.
  - `bwareaopen`: Removing connected components with fewer than 100 pixels.

Object Validation and ROI Extraction

Labeling and Component Analysis:

- `bwlabel`: Labeling the mask based on connected components, assigning a unique number to each object.
- `regionprops`: Calculating properties (bounding box, area, image) of each object.

Validation Criteria:

- **Aspect Ratio:** Between $1/1.9$ and 1.9 (calculated as the width-to-height ratio of the bounding box).
- **ROI Area:** Between $\frac{w \cdot h}{25}$ and $\frac{w \cdot h}{3}$, where w and h are the width and height of the image.

ROI Extraction:

- The binary ROI is extracted using the `regionprops.Image` property.

Shape Classification

This section illustrates the strategies for classifying ROIs based on their shape, distinguishing between **circles** and **triangles**.

Objectives:

- Define and compare the existing approach (Ellahyani) with the proposed methodology.
- Extract relevant geometric features and use an SVM classifier.

Approaches Comparison

Ellahyani's Approach:

- Use of Hu invariant moments on binary reference patches.
- Calculation of Euclidean distance between reference patches and ROIs.
- *Limitation:* High variability of Hu moments across images of the same sign.

Proposed Methodology:

- Combination of Hu moments and circularity measure (obtained via `regionprops`).
- Creation of an 8-element feature vector for each ROI.
- Use of the GTSRB dataset for training the classifier.

Implementation Pipeline

Main Steps:

1. **Segmentation:** Extraction of binary ROIs through color segmentation.
2. **Feature Extraction:** Calculation of the 7 Hu invariant moments and circularity for each ROI.
3. **Classification:** Training of an SVM model and integration into the pipeline.

Dataset: GTSRB

- Selection of signs with circular and triangular shapes (red and blue colors).
- Filtering based on *aspect ratio* (between 1/1.9 and 1.9).
- Balancing process via undersampling:
  - From 39,209 initial images, 18,783 were processed (11,986 circles and 6,797 triangles).
  - Final balanced dataset with 6,797 samples per class (total 13,594).

Feature Extraction for SVM

Extracted Features:

- **Hu Moments:** The 7 invariant moments that capture the geometric properties of the shape.
- **Circularity:** Measure obtained from `regionprops` quantifying how circular a shape is.

Feature Vector:

- Concatenation of the 7 Hu moments and circularity $\rightarrow$ 8-element vector for each ROI.

These features are used to train an SVM model capable of distinguishing between **Circle** and **Triangle**.

SVM Classification Model

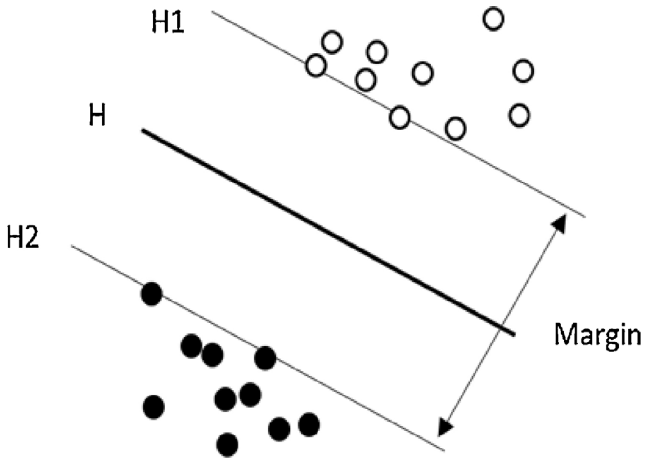
Key Characteristics:

- Supervised algorithm for classification and regression.
- Determines the optimal hyperplane that separates classes by maximizing the margin between them.
- Uses the **kernel trick** to handle non-linearly separable data.

Application:

- Classification of ROIs into two classes: **Circle** and **Triangle**.
- The 8-dimensional feature vectors (7 Hu moments + circularity) serve as input for training.

Binary Classification with SVM



Example of Feature Extraction

h_1	h_2	h_3	h_4	h_5	h_6	h_7	**Circ.**
0.7930	3.7416	6.1562	7.8064	-14.7600	-10.0358	15.1257	0.5664

h_1	h_2	h_3	h_4	h_5	h_6	h_7	**Circ.**
0.7242	4.3796	2.4168	5.7990	10.4917	-8.5937	9.8288	0.5277

Optimization of the SVM Model

Procedure:

- **Grid Search** with cross-validation for parameter selection.
- Parameters tested:
  - **Kernel:** `linear`, `rbf`, `polynomial`
  - **BoxConstraint:** 1, 3, 5

Optimal Parameters Found:

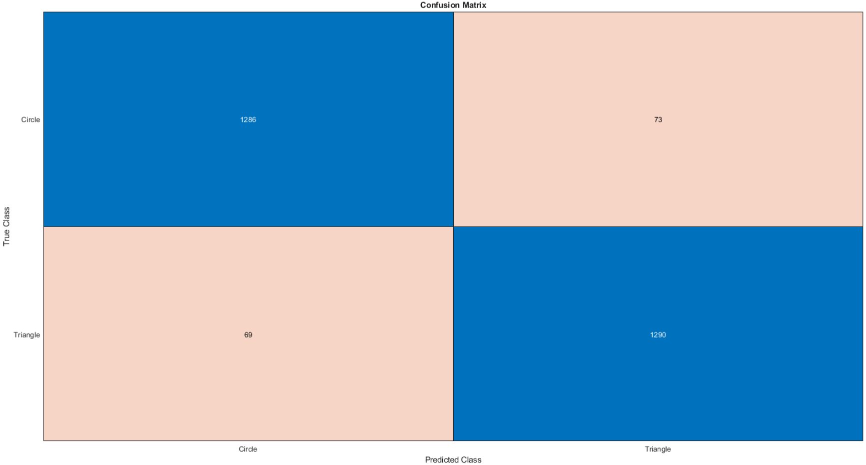
- **Kernel:** RBF
- **BoxConstraint:** 3.0

Results:

- **Training Set** (10,876 samples): 94.78% accuracy.
- **Test Set** (2,718 samples): 94.35% accuracy.

Evaluation of the SVM Model

Confusion Matrix on Test Set



Integration of the SVM Model into the Pipeline

- The SVM model, trained with the 8-feature vector, is integrated into the main pipeline.
- Classification of ROIs based on shape: **Circle** or **Triangle**.
- ROIs with confidence below a predefined threshold are discarded.
- Correctly classified ROIs are saved in RGB format for further signal recognition steps.

Traffic Sign Recognition

The third and final step of the work focuses on traffic sign recognition.

Starting from the segmented ROI obtained in the previous two steps, recognition proceeds through feature extraction.

Feature Extraction: HSI-HOG+LSS

The extracted features are combined to form a descriptor called **HSI-HOG+LSS**.

Main Steps:

1. Calculation of HOG features for each channel of the HSI image.
2. Calculation of LSS features to capture structural similarity.
3. Concatenation of HOG and LSS vectors to obtain the final descriptor.

HOG Features

HOG features are calculated for each channel of the HSI image instead of using a grayscale image.

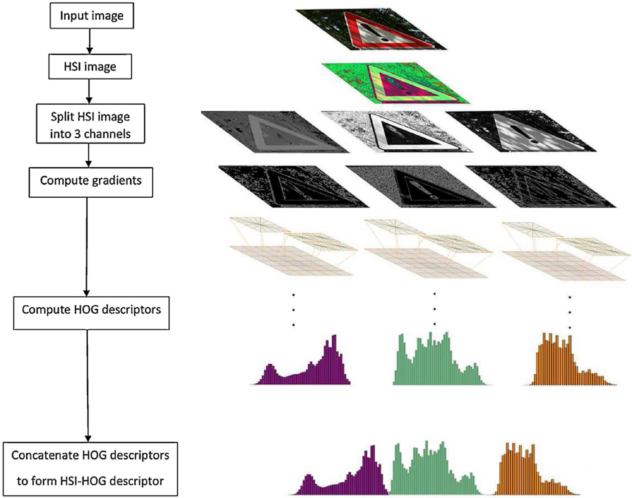
Key Idea: The distribution of intensity gradients characterizes the local appearance and shape of an object well.

Parameters Used (image normalized to 40 $\times$ 40):

- **Block Size:** 8×8
- **Block Overlap:** 4×4
- **Cell Size:** 5×5
- **Number of Bins:** 9

Result: Three HOG vectors of size 576, one for each HSI channel.

HSI-HOG Extraction



LSS Features

LSS features capture structural self-similarity by correlating smaller cells of the image with a central patch.

Procedure:

- A correlation surface is computed, normalized, and projected into a space defined by angular and radial intervals.
- The maximum value for each interval is taken as a feature.

Parameters Used:

- **Patch Size:** 3×3
- **Window Radius:** 10
- **Number of Angular Intervals:** 20
- **Number of Radial Intervals:** 4

Result: Log-polar coordinates divided into 80 bins (20×4), each contributing one feature.

HSI-HOG+LSS Descriptor

Descriptor Construction:

- Compute the HOG descriptor for each channel (Hue, Saturation, Intensity) on the normalized ROI (40×40).
- Concatenate the three HOG vectors with the LSS feature vector.

The resulting vector (1808 total features), named **HSI-HOG+LSS**, is then used as input for the subsequent classification step.

Dataset Balancing

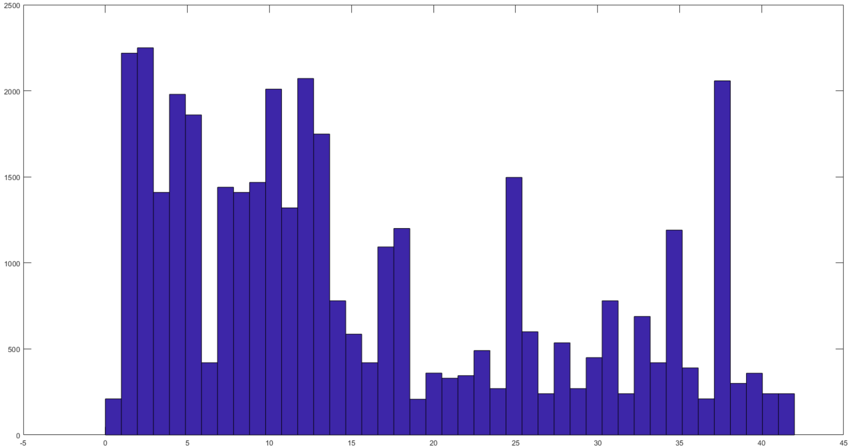
Issues Identified:

- **Class Skewness**: Non-uniform distribution across the 43 classes.
- **Bias in ML Models**:
  - Over-represented classes tend to dominate training.
  - Poor recognition of minority classes.
- **Hardware Limitations**:
  - Computationally expensive training on the full dataset.
  - Constraints due to available computing power.

Solution Adopted:

- **Undersampling**: Uniform selection of 208 examples per class.
- **Balanced Dataset**: $43 \times 208 =$ **8,944 images**.
- Reduction of the original dataset by **78%**, facilitating training and mitigating bias.

Original Class Histogram



Random Forest Classifier

Key Characteristics:

- **Ensemble of Trees:** The model consists of numerous decision trees, each trained on a random sample (bootstrap) of the data.
- **Random Feature Selection:** At each node, the best split is chosen from a random subset of predictors, introducing additional variability.
- **Robustness to Noise and Overfitting:** The ensemble approach improves accuracy and generalization compared to individual decision trees.

Key Model Parameters:

- **Number of Trees**
- **Minimum Leaf Size (MinLeafSize)**
- **Number of Features per Split**

Training Strategy

Objective:

- Find the optimal number of trees.

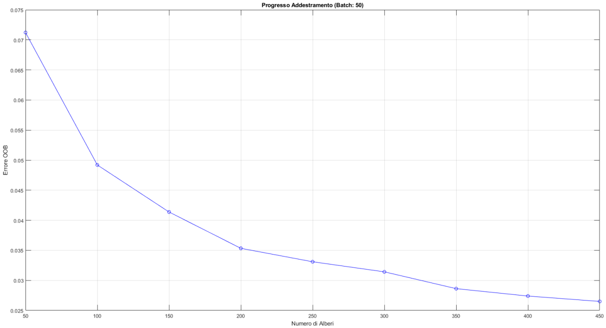
Training Procedure:

- **Incremental Growth:** The model is trained in batches (50 trees per iteration).
- **Out-Of-Bag (OOB) Error:** The OOB error is calculated at each iteration, estimating accuracy on data not used for training each tree.
- **Convergence Criterion:** If, over a window of consecutive iterations, the change in OOB error falls below a predefined threshold, the optimal number of trees is fixed.

Evaluation of OOB Error

Results:

- Convergence reached at **400 trees**.
- Training completed in approximately 2.41 minutes.



(Behavior of the OOB error as the number of trees increases)

Evaluation of the Random Forest Model

Test Set Balancing:

- Same procedure as adopted for the training set.
- Reduction of the set from 12,630 to 2,537 images.

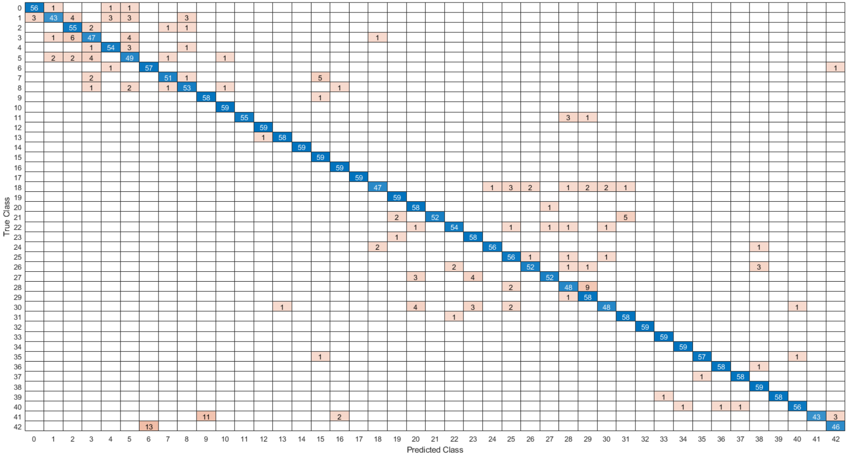
Metrics Obtained:

Metric	Value (%)
Accuracy	92.94
Precision	93.24
Recall	92.94
F1-score	92.87
Specificity	99.83

Table: Results obtained on the balanced test set.

Note: Prediction completed in 2.52 seconds.

Confusion Matrix of the Random Forest Model



Confusion Matrix for traffic sign classification.

General Pipeline Workflow

The traffic sign recognition pipeline consists of the following steps:

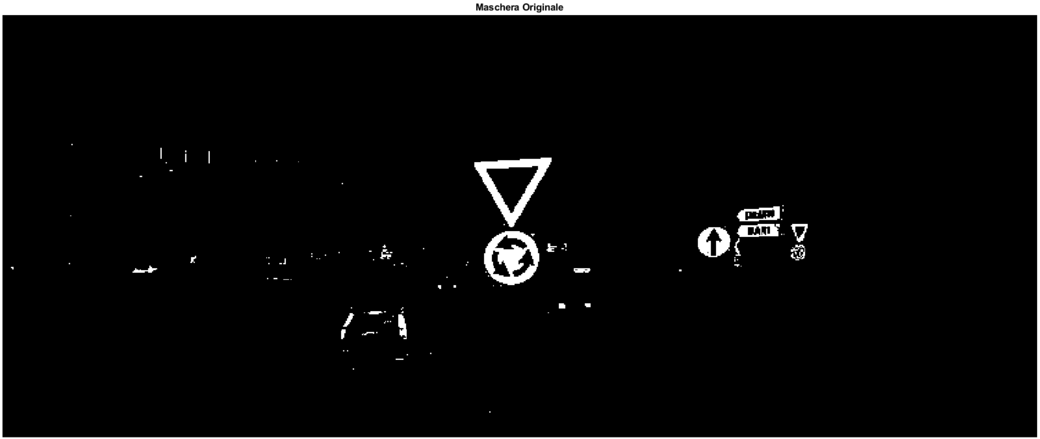
1. **Image Acquisition:** Capture of the image to be analyzed.
2. **Color Segmentation:** Creation of binary masks to isolate signs.
3. **ROI Extraction:** Detection and validation of regions of interest.
4. **Shape Classification:** Identification of the shape (circle, triangle) of ROIs.
5. **Feature Extraction:** Computation of relevant features for sign recognition.
6. **Final Prediction:** Final classification of the traffic sign.

Image Acquisition

Immagine Input

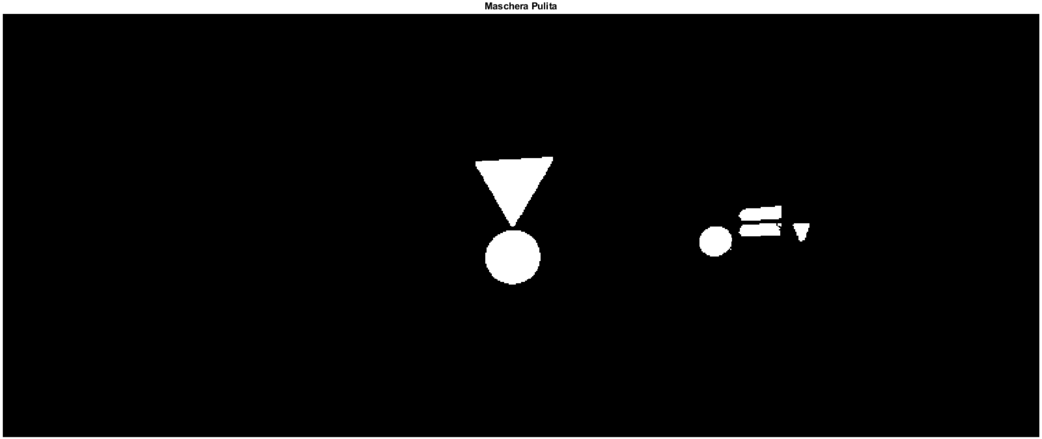
Description: The input image is acquired from a source (e.g., camera or dataset) for processing.

Color Segmentation and Initial Mask



Initial Mask

Color Segmentation and Cleaned Mask



Cleaned Mask

Extraction of Valid ROIs

Oggetti Rilevati

Description: Valid regions of interest (ROIs) are identified from the cleaned mask based on validity criteria (aspect ratio, area).

Shape Classification

Rilevamento Forme

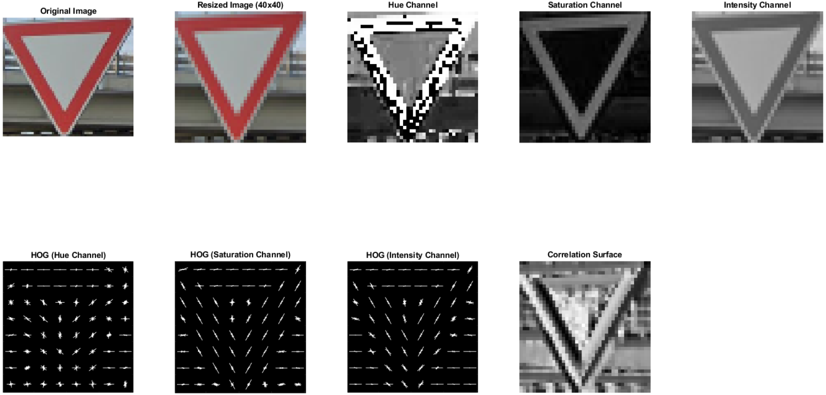
Description: Valid ROIs are analyzed to determine their shape (circle or triangle) using an SVM model or discarded if not valid.

Extracted ROIs

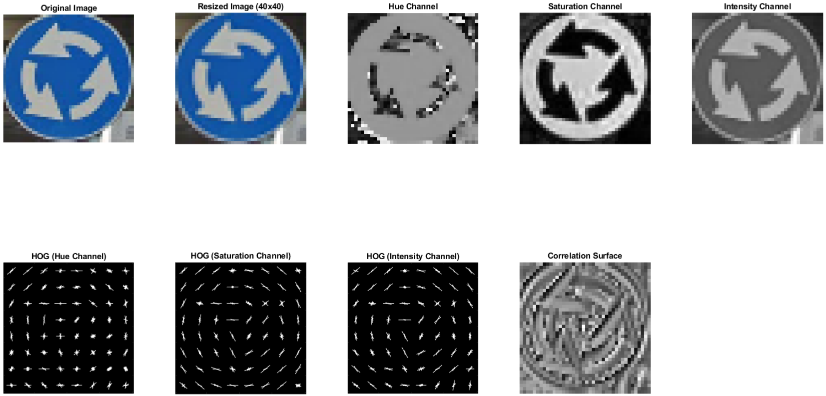
ROI Estratte

Description: Correctly classified ROIs are extracted and prepared for the subsequent signal recognition phase.

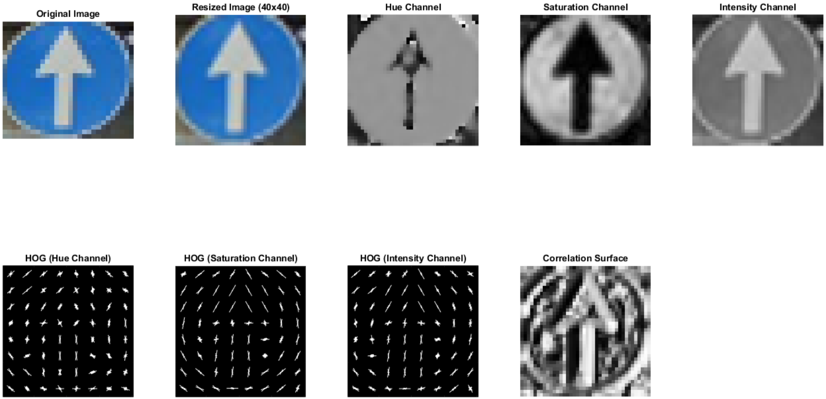
Feature Extraction - ROI 1



Feature Extraction - ROI 2



Feature Extraction - ROI 3



Final Prediction

Conclusions

- **Integrated Approach:**
  - A complete pipeline has been developed that integrates color segmentation, shape classification, and traffic sign recognition.
- **Effective Methodologies:**
  - The use of geometric features (Hu moments and circularity) and HSI-HOG+LSS descriptors effectively characterized the signs.
  - Training of SVM and Random Forest models ensured accuracy and robustness.
- **Significant Results:**
  - Excellent results in terms of accuracy, precision, recall, and F1-score, both in shape classification and traffic sign recognition.
  - Validation on real datasets (GTSRB and images from Google Maps Street View) confirmed the pipeline's effectiveness.

Weak Points

- **Segmentation under Low Lighting Conditions:**
  - Low illumination can compromise the correct identification of pixels of interest.
- **Partial Occlusion of Signs:**
  - Signs partially obscured or covered by obstacles reduce the accuracy of segmentation and classification.
- **Manual Threshold Setting:**
  - The need to manually define shape classification thresholds can lead to errors in the presence of environmental variations.
- **Generalization in Complex Scenarios:**
  - Complex backgrounds or non-standard signs may require further optimizations to improve system robustness.

Future Work

- **Model Improvement:**
  - Experiment with other feature extraction and classification techniques.
  - Optimize hyperparameters and explore hybrid models.
- **Integration into ADAS Systems:**
  - Evaluate the integration of the pipeline into autonomous or assisted driving systems.
  - Extend recognition to other types of signs and environmental conditions.
- **Dataset Expansion:**
  - Expand the dataset to include a greater variety of signs and scenarios.

Citations

This work is inspired by the method proposed by Ellahyani et al. [1].
The theoretical foundation for digital image processing is provided by Gonzalez and Woods [2].

References

[1] Ayoub Ellahyani, Mohamed Ansari, and Ilyas El jaafari. "Traffic sign detection and recognition based on random forests, Applied Soft Computing". In: *Applied Soft Computing* 46 (Feb. 2016). DOI: 10.1016/j.asoc.2015.12.041.

[2] Rafael C. Gonzalez and Richard E. Woods. *Digital image processing*. New York: Pearson Education, 2018, p. 1022.